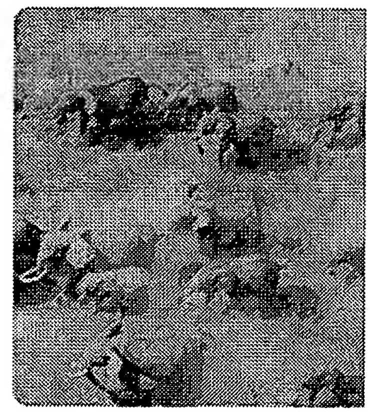
BEST AVAILABLE COPY



TechEncyclopedia

More than 20,000 IT terms

Results found for: **channel**

channel

Search TechWeb For:

- channel

(1) A high-speed metal or optical fiber pathway between the computer and the control units of the peripheral devices. Channels are used in mainframes and high-end machines. Each channel is an independent unit that can transfer data concurrently with other channels as well as the CPU. For example, in a 10-channel computer, 10 streams of data are being transmitted to and from the CPU at the same time. In contrast, the bus in a personal computer serves as a common, shared channel between all devices. Each device must wait for its turn on the bus.

(2) In communications, any pathway between two computers or terminals. It may refer to the physical medium, such as coaxial cable, or to a specific carrier frequency (subchannel) within a larger channel or wireless medium.

(3) Information on a particular subject that is transmitted into the user's computer from a Webcast site via the user's browser or push client. It is the Internet's counterpart of the TV or radio channel. See Webcast, push client and push technology.

(4) The distributor/dealer sales channel. Vendors that sell in the channel rely on the sales ability of their dealers and the customer relationships they have built up over the years. Such vendors may or may not compete with the channel by selling direct to customers via mail order.

TERMS SIMILAR TO YOUR ENTRY

Entries before channel

- challenge/response
- change control
- change file
- change management
- change request

Entries after channel

- channel attached
- channel bank
- channel bonding
- channel coding
- Channel Definition Format

DEFINE ANOTHER IT TERM

Define

Or get a random definition

comparing a number of counted quiet slots with the arbitration count number for the node; and means, responsive to the number of counted quiet slots equaling the arbitration count number, for beginning transmission of a message on the bus. Also, a means, responsive to a node winning an arbitration, changes the arbitration count number of each node of the plurality of nodes.

Other and further aspects of the present invention will become apparent during the course of the following description and by reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings, in which like numerals represent like parts in the several views:

FIG. 1 is a logical block diagram of a computer system using the invention.

FIG. 2 is a physical block diagram of a computer system using the invention.

FIG. 3 is a flow chart showing one embodiment of the invention.

FIG. 4, comprising 4A, 4B and 4C is a state diagram of an arbitration apparatus using the invention.

FIG. 5 is a state diagram of an arbitration apparatus having quiet slots 0 and 1 reserved.

FIG. 6 is a block diagram showing two nodes attempting to simultaneously transmit and so placing corrupted data on the bus.

DETAILED DESCRIPTION

FIG. 1 is a logical block diagram of a computer system 20 having a bus 100, and nodes 110A, 110B through 110N. Nodes 110A, . . . 110N communicate through communication signals transmitted on bus 100.

FIG. 2 is a physical block diagram of computer system 20 and bus 100. Connections 100A, 100B, through 100N connect their respective nodes to a star coupler 120. The star coupler provides communication interconnection between connections 100A, . . . 100N, thereby providing a pathway for nodes 110A, . . . 100N, to communicate. For example, bus 100 may be an electrical co-axial cable, and star coupler 120 may be an impedance matching network for interconnecting the co-axial cables. In the event that a first node, say node 110A, desires to transmit a message to another node, say for example node 110B, then node 110A transmits its message through bus 100 to all nodes connected to bus 100. The message contains an address which only node 110B recognizes. Accordingly, node 110B internally adjusts itself to receive the message. Other nodes may receive all of the message or part of the message, depending upon the protocol adopted at the node.

A problem in communications between nodes, is that all nodes transmit on the same bus 100, and so if two nodes begin their transmission at the same time, their transmissions will become garbled on the bus. A data "collision" occurs when two nodes transmit messages simultaneously. A process of arbitration for the bus must be adopted for dealing with data collisions on the bus.

A common solution to the data collision problem that has been adopted for the ETHERNET communications pathway is for each node to look for a "carrier" signal on the bus before it begins transmission.

The ETHERNET protocol is set out in IEEE Standard 802.3 and the ISO CSMA/CD Standard (CSMA/CD stands for Carrier Sense Multiple Access/-Collision Detection).

Under the ETHERNET protocol, in the event that the node finds no carrier on the bus, it then begins transmission.

Also under the ETHERNET protocol, in the event that the node detects another node's carrier on the bus, a number of strategies may be adopted to reduce data collision and congestion on the bus. For example, a node may begin transmitting immediately upon sensing that no carrier is present on the bus. Alternatively, a node may wait a random amount of time before beginning a transmission after first detecting that there is no carrier on the bus. However, these solutions may lead to data collisions on the bus because two nodes may test the bus at substantially the same time, and then may begin transmitting their messages at substantially the same time without realizing that the other node is transmitting. However, a node is capable of detecting the fact that a data collision is occurring.

After a node detects that a data collision is occurring, then the node takes corrective action depending upon the protocol that is adopted. For example, a node, after detecting a data collision, may use the exponential backoff protocol. In the exponential backoff protocol, a node waits a random amount of time before attempting to re-transmit the message that was ruined by the data collision. And if a second data collision occurs, the nodes involved then double the time interval from which the random time is chosen. If subsequent collisions occur, then the nodes keep doubling the time interval from which the random time is chosen until no data collision occurs.

The invention is an arbitration method which permits the nodes to communicate on a bus 100 with a very small probability of data collision. FIG. 3 is a flow diagram showing the invention. Each node is assigned a node number represented by the letter I. Additionally, each node is assigned an arbitration count number C(I), where the (I) indicates that the arbitration count number is assigned to node number I.

The computer system including bus 100 and nodes 110A, 110B, . . . 110N must be initialized before operation begins. At initialization the nodes are each assigned an initial arbitration count number C(I). As shown in FIG. 3, at initialization a system begins at ENTER block 120. The system immediately proceeds to initialization block 130 wherein the arbitration count number C(I) for each node is initially assigned the value of I+1. That is, node No. 1 is assigned arbitration count number 2, node No. 2 is assigned arbitration count number 3, . . . etc.

The node then proceeds to block 135 where the value of the "quiet slot" counter K is initially set to zero. K will hold the value of the number of quiet slots elapsed since carrier was last detected on the bus.

A quiet slot is a predetermined length of time. The node is required to count a number of quiet slots equal to the arbitration count number of the node before the node can begin transmitting. If the count is interrupted by the node's detecting the carrier from another node, then the node does not transmit.

The time length of a "quiet slot" is selected on the basis of the speed of interface electronics in the nodes and the length of time required for an electrical signal to propagate from a first node to a second node where the nodes are separated by the maximum length of the bus. For example, if the separation between the most widely separated nodes is 100 meters and the speed of an electrical pulse on the bus is approximately 0.2 meters per

nanosecond, then a quiet slot may be chosen to be approximately 500 nanoseconds plus the response time of the interface electronics used in each node.

In an embodiment of the apparatus incorporating the Strecker, et al. disclosures and sold by the assignee of this patent, a star coupler physical arrangement as shown in FIG. 2 is used, all connection cables 100A . . . 100N are cut to a defined length of 45 feet, and a quiet slot is chosen to be 1.14 microseconds. The value of 1.14 microseconds takes into account both the propagation time as illustrated above, and the speed of the interface electronics.

After block 135 the system proceeds to block 155 where the node waits for one quiet slot and updates the count K to K+1. After this the system proceeds to block 157 where it tests the bus for the presence of a carrier.

In the event that a carrier is not detected at block 157, the system proceeds to block 160 wherein a test is made to determine if it is this node's turn to transmit. This test is done by comparing the number of quiet slots counted, K modulo M against the arbitration count number C(I). If the two are equal then the node assumes that it has the turn to transmit, and proceeds to block 140 to check if it has a transmit request. In the event that K modulo M does not equal the arbitration count C(I), the node returns to block 155 and continues to count quiet slots.

At block 140 if the system detects a transmit request, it proceeds to block 180 where it begins transmitting its message on bus 100. Following completion of transmission, the node resets its arbitration count to M, as shown in block 215, and returns to block 135 to again start the quiet slot count.

At block 140 if no transmit request is detected, the system moves to block 150 wherein an elapsed time counter is tested to determine if the time elapsed, since a transmit request was received, exceeds a "reset time". The reset time is a predetermined value. In the event that the time interval has exceeded the reset time, then the system is reinitialized along path 151. In a practical system, the reinitialization may be achieved by the node transmitting a "reset message" over the bus. Since each node receives all messages transmitted on the bus, each node in the system will be able to reset its arbitration count on receipt of the reset message.

In the event that at block 157 a carrier is detected, the system proceeds to block 190 where a determination of the node number of the transmitting node is made. This determination is possible because each node receives all messages transmitted by all nodes, and a protocol adopted by the system requires that a header portion of a message contain the address of the transmitting node. Each node can interpret the header produced by the transmitting node to make the determination, shown in block 190, of the node number of the transmitting node.

The system then proceeds to block 200 where the arbitration count number of each node is updated. Block 200A shows the first alternative embodiment of updating the arbitration count number. Block 200B shows a second alternative embodiment of updating the arbitration count number.

After block 200, the system proceeds to block 210 wherein the nodes go into a wait state until the carrier on the bus ceases, following which the system returns to block 135 and arbitration starts again.

First Alternative Embodiment

FIG. 4A, FIG. 4B, and FIG. 4C show an example of the invention. In the example, it is assumed that there are 16 nodes in the system, and node numbers are assigned 0–15. The figures show a sequence of events, and those events represent transmission by various nodes onto bus 100. Suppose that nodes 4, 6, 9 and 15 have transmission requests. FIG. 4A shows the nodes as represented by numbers 250A, where numbers 250A are indicated with an I. The nodes are numbered 0, 1, 2, 3, . . . 15. That is, block 140 as shown in the flow diagram of FIG. 3 will answer "yes" for each of these nodes.

As shown in block 200A, a first alternative method of updating the arbitration count number is for each node to use the number of quiet slots counted by the winning node. It is easy for each node to use the number of quiet slots counted by the winning node because each node knows how many quiet slots it counted before losing the arbitration to the winning node. Each node then subtracts the number of counted quiet slots from its arbitration count number C(I), modulo M, in order to obtain its arbitration count number. The arithmetic may be expressed by the modulo M equation:

$$C(I) = \left(C(I) - \begin{pmatrix}\text{Number of Quiet Slots}\\ \text{Counted by the}\\ \text{Winning Node}\end{pmatrix}\right) \quad \text{Modulo } M$$

if C(I) is not equal to the number of quiet slots counted by the winning node. and,

$$C(I) = M$$

if C(I) is equal to the number of quiet slots counted by the winning node.

The nodes are initialized by each node being assigned an arbitration count number equal to its node number plus one.

FIG. 4A, FIG. 4B, FIG. 4C illustrate the process of nodes counting the number of quiet slots counted by the winning node, in order for each node to compute its new arbitration count number. In the example shown in FIG. 4A . . . FIG. 4C, suppose nodes 4, 6, 9, and 15 have pending transmission requests. Node 4 transmits first. After node 4 transmits, the assignment of arbitration count numbers is as shown in FIG. 4A. Node 5 has arbitration count number equal 1 (C(5)=5−4=1), node 6 has its arbitration count number equal 2 (C(6)=6−4=2), and so forth through node 15 which has arbitration count number (C(15)=15−4=11) 11, and node 3 has arbitration count number (C3)=(3−4) modulo 16=15) 15. Accordingly, after node 4 finishes its transmission, each node detects the absence of carrier on bus 100, and so each node begins counting quiet slots. Node 5 counts 1 quiet slot, and since node 5 has no pending transmission request, node 5 passes the opportunity to transmit on bus 100. Node 6 counts the first quiet slot and also a second quiet slot, node 6 has a pending transmission request, and so upon counting quiet slot number 2, node 6 begins transmission on bus 100. Also node 9 counted 2 quiet slots before losing the arbitration to node 6, and node 15 also counted 2 quiet slots before losing the arbitration to node 6. Accordingly, node 9 subtracts 2 from its present arbitration count number f 5 in order to compute a new arbitration count number of 3, and node 15 subtracts 2 quiet